Projet TSI

Reconnaissance de visages par eigenfaces

Sommaire

Introduction : En quoi consiste le projet ? Quel est son intérêt ? Comment allons-nous procéder ? Quelles technologies utiliser ?

Plan d'étude :

A) Point de départ: Nous travaillons sur une base de données de référence

B) Analyse en composantes principales

C) Projection dans le sous-espace des visages

D) Identification d'un visage à partir du numéro *id*

E) Comment distinguer une image visage d'une image quelconque?

F) Recalage des visages

Implémentation du système complet : Présentation

Bilan: Quelles améliorations pourrait-on apporter ?

Annexe : Mode d'emploi des différents programmes 'MIX.m' pour l'étude préalable du système

INTRODUCTION :

Reconnaître une personne est directement impliqué dans une multitude de secteurs d'activités. L'être humain possède un pouvoir de reconnaissance de visages, qui se révèle néanmoins insuffisante pour certains cas. Nous essayons ainsi de mettre en place des systèmes automatiques pour remplir la mission, notamment dans le domaine de sécurité, où il est indispensable d'utiliser un système performant. Le visage est, dans un cas général, un bon facteur de distinction entre les êtres humains.

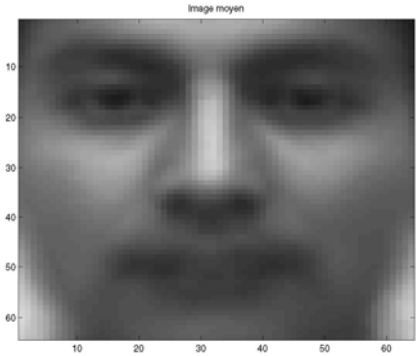
Le but de ce projet sera donc de développer un système qui manipule les images représentant des visages humains. Il doit notamment nous dire si une image est effectivement un visage humain, et si oui, on doit pouvoir effectuer un recalage manuel de l'image pour focaliser sur le visage (en prenant comme référence la position des yeux) afin de faciliter la reconnaissance avec ce système, enfin le reconnaître à partir d'une base de données de visages de référence. Pour cela nous allons utiliser la méthode de reconnaissance par ***eigenfaces***.

Nous allons donc partir d'une base de données de visages recalés pour optimiser notre système de reconnaissance. Dans un premier temps, nous allons étudier les bases théoriques, à savoir le principe de l'analyse en composantes principales pour introduire les éléments d' eigenfaces. Ensuite, nous étudierons la reconstruction des images par projection dans le sous espace des visages, le paramètre d'étude primordial K sera introduit et sa valeur optimale sera fixée entre un certain intervalle après une étude d'erreur. Ceci étant fait, on peut enfin procéder aux étapes d'identification et de reconnaissances proprement parlées.

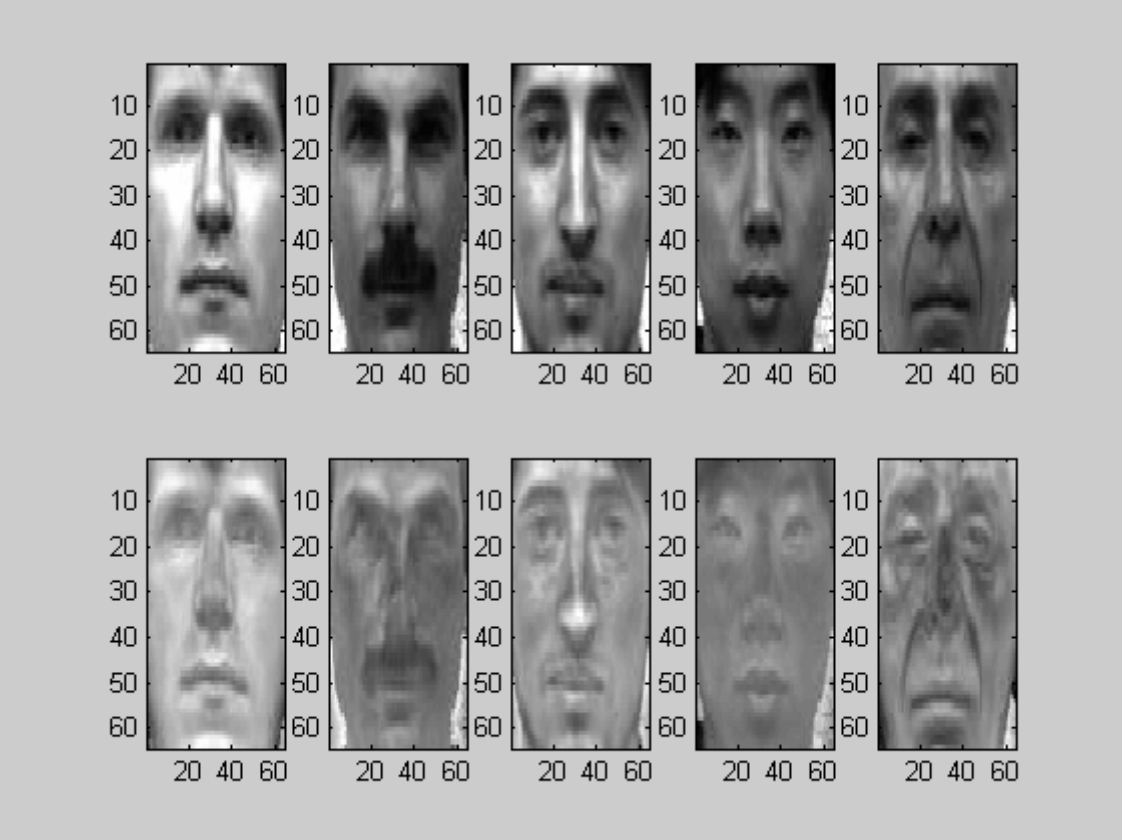
Enfin, le système complet sera implémenté, fournissant aux utilisateurs une interface pour tester le résultat de notre travail. Les fonctions sont toutes implémentées sous **Matlab 7.0.**

BASES DE DONNEES DE VISAGES :

Nous utiliserons la matrice I pour stocker la base des images de référence. Elle est de taille n*m (4096*7), et comporte des réels au format double, donc prendra 8*n*m=8*4096*105= 3.44Mo.On peut afficher le visage moyen :



Puis les cinq premiers visages et accompagnés des visages centrés correspondants :



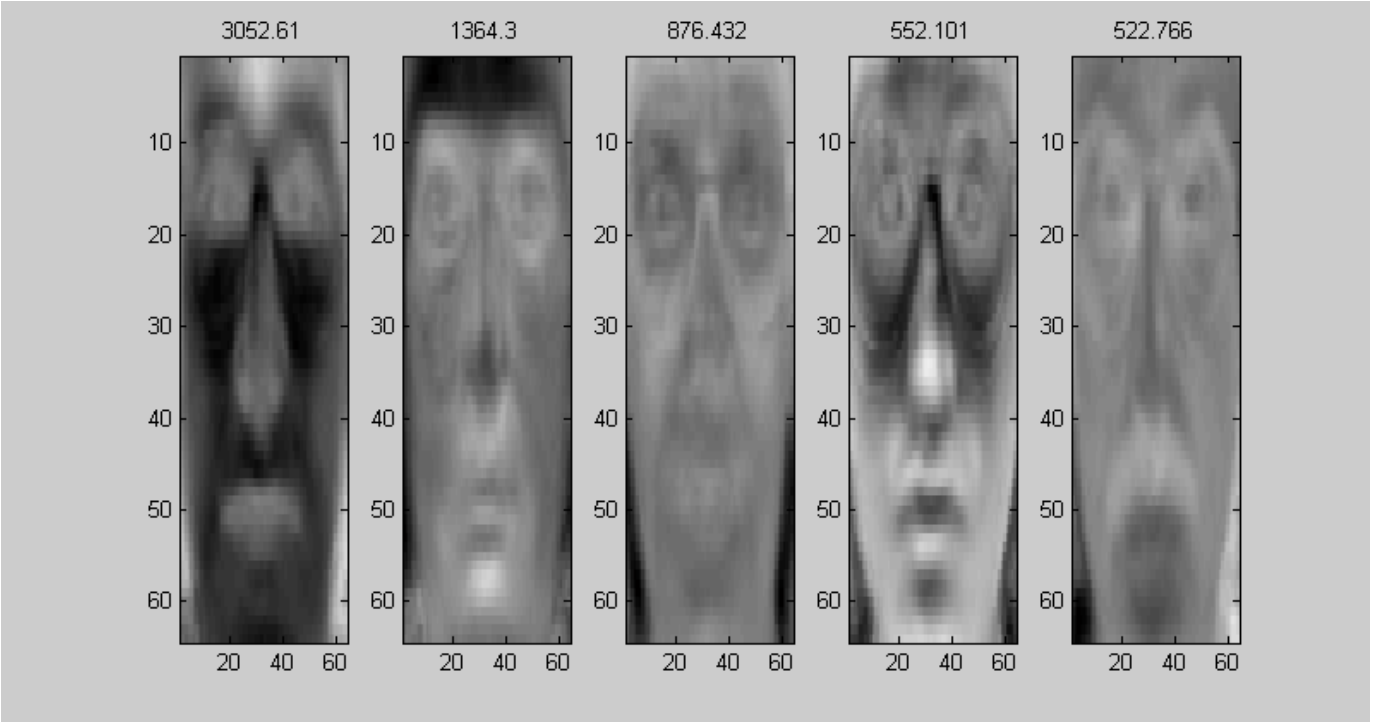
Le premier constat est le manque de clarté des visages centrés, ceci s'explique par le fait que des informations ont été enlevées par rapport aux visages d'origine.

ANALYSE EN COMPOSANTES PRINCIPALES (ACP) :

Sachant que nous procédons par une approche image, chaque image de visage est considérée comme un vecteur dans un espace multidimensionnel (autant que le nombre de pixels dans l'image).

Nous définissons la matrice A comme étant la matrice représentant des visages centrés, les eigenfaces sont ainsi les vecteurs propres de la matrice de covariance A.A'. Nous appliquons la décomposition en valeurs simples (DVS) : $A=U.S.V_T$. En mémoire, U prendra 8*n*n octets, V 8*m*m octets et S 8*n*m octets. Nous préférons la DVS à la décomposition en valeurs propres car elle présente une réduction en terme de coût en mémoire.

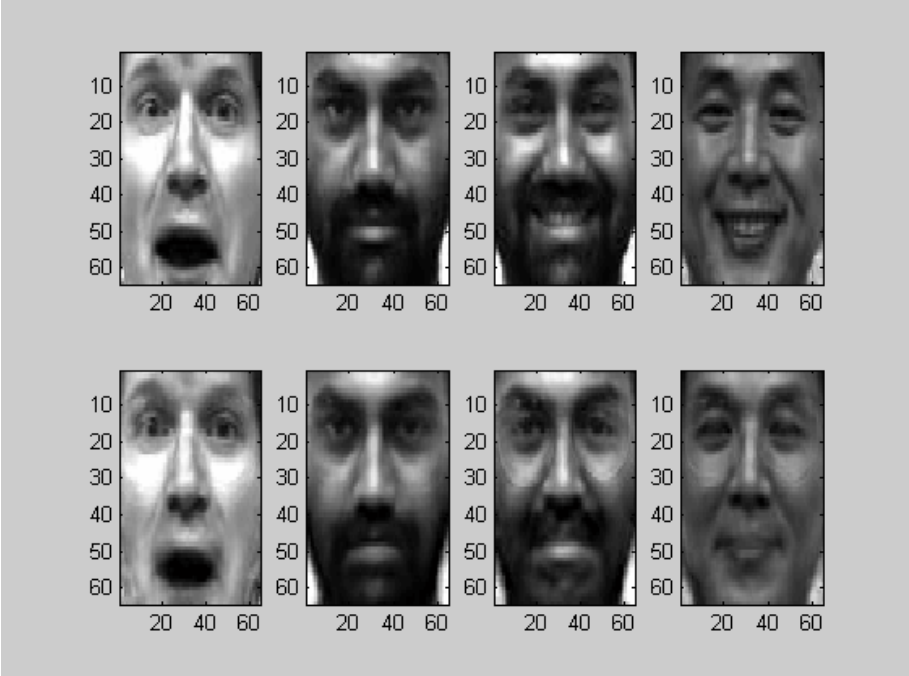
On peut afficher les cinq premières eigenfaces avec leur valeur propre respective en en-tête:



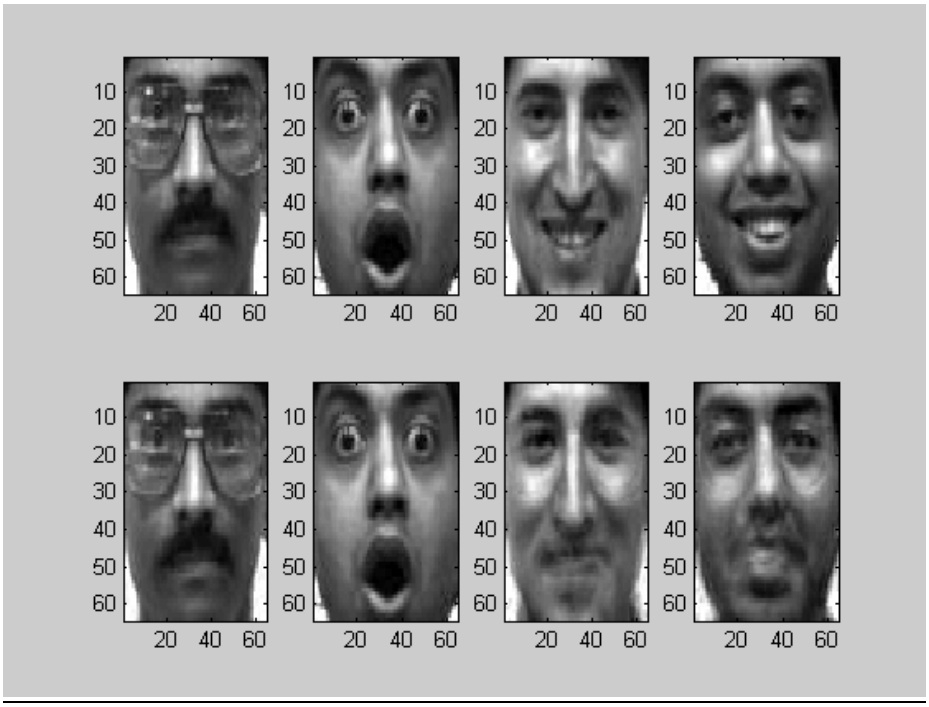
Par la suite, nous allons réaliser différentes expériences pour déterminer les valeurs optimales des paramètres qui permettent de réaliser les reconnaissances, en commençant par le nombre de eigenfaces à prendre.

PROJECTIONS DANS LE SOUS ESPACES DES VISAGES :

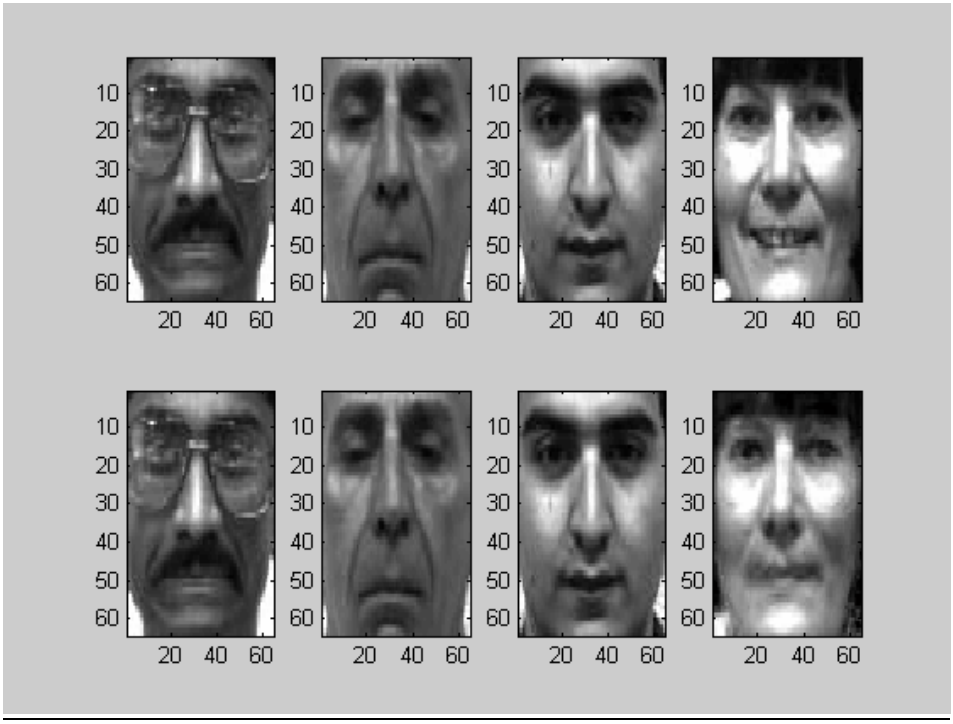
K=30



K=80

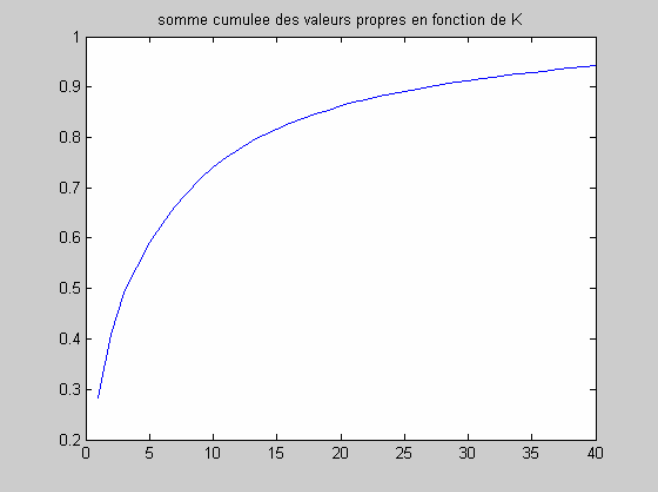


K=m



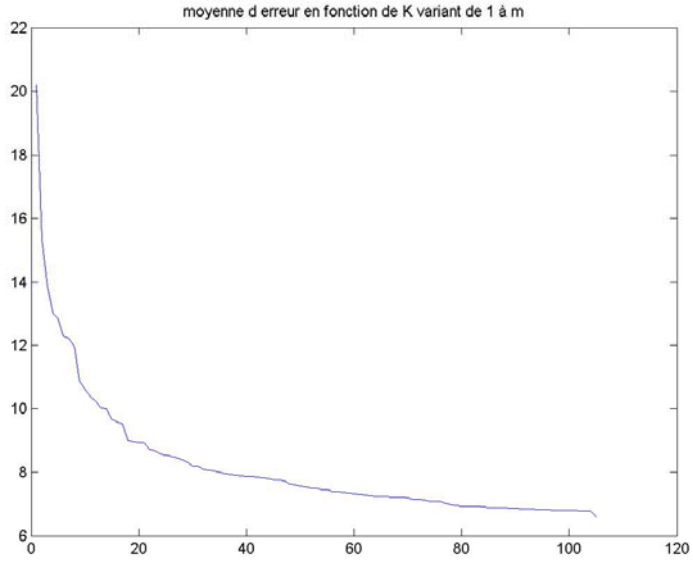
On constate qu'il n'y a presque pas d'erreur de reconstruction pour K=m, les visages reconstruits correspondent bien aux visages d'origine, les petits décalages viennent du fait que toutes les informations n'ont pas été rassemblées avec les eigenfaces.

On normalise les valeurs propres pour que leur somme soit égal à 1 et on trace la courbe de leur somme cumulée pour les K (K=40) premières eigenfaces.



D'après ce graphe, si on prend 0,9 comme référence de bonne reconstruction, il faudrait alors prendre une valeur de K minimale de 30.

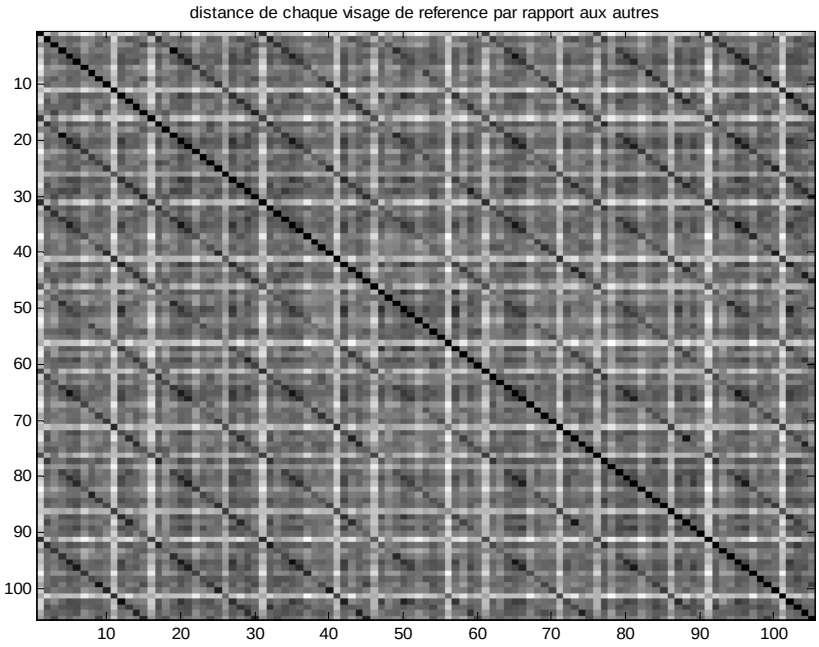
Ensuite on peut tracer la courbe illustrant l'évolution de la moyenne d'erreur de reconstruction des visages de test. Celui-ci reste cohérent avec le graphe précédent, l'erreur est inférieure à 8 pour K supérieur à 30.



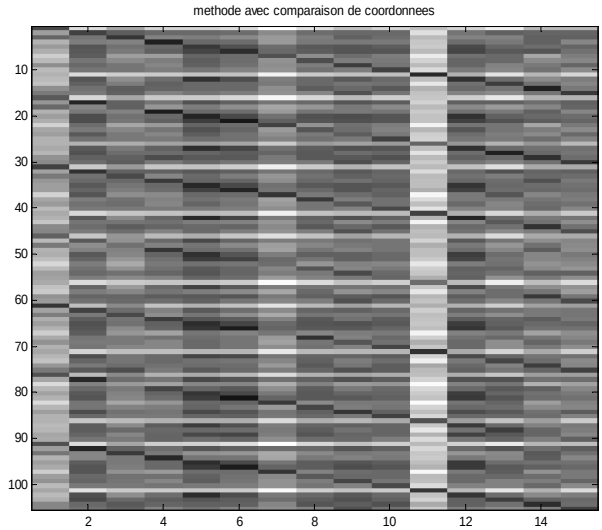
IDENTIFICATION :

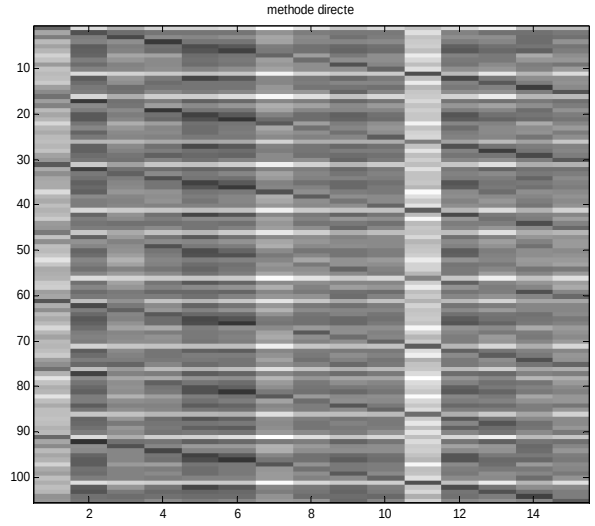
On peut estimer la distance dans le sous-espace pour diminuer le nombre de calcul à faire.
En prenant K=20, on visualise dans le schéma ci-dessous, sous forme d'image matricielle, la distance de chaque visage de référence par rapport à tous les visages dans le sous-espace. On trouve pour deux visages de même classe des distances maximales et minimales de 18 et 0, et pour les visages de classes différentes, ces valeurs sont respectivement de 25 et 3.5. Ainsi en prenant par exemple une valeur de

M=2, on n'obtiendra qu'un résultat satisfaisant seulement pour les visages très proches et pour M=20, la marge d'erreur sera plus grande pour les visages de même classe.



Maintenant on va comparer la méthode qu'on utilise avec les coordonnées avec un méthode où l'on procèderait par une méthode directe. Les graphes ci-dessous montrent la différence: on trouve 3 différences de reconnaissance dont les positions sont indiquées par le vecteur Pos.



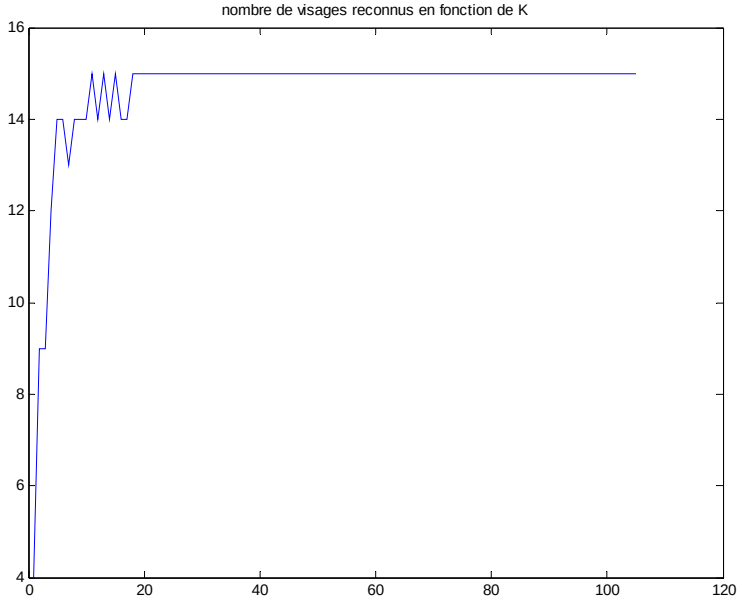


Pos=	61	92	3	19	20	81	37	68	84	25	11	42	28	14	15
	61	92	3	19	35	66	37	68	9	25	11	42	28	14	15

On remarque que le taux d'identification est le même pour les 2 alors que l'erreur d'identification est plus grande pour la méthode directe (ça se voit au niveau des matrices du fait que celle de gauche est plus clair et donc les valeurs sont plus proches de 1 après les avoir normalisées et donc sont plus

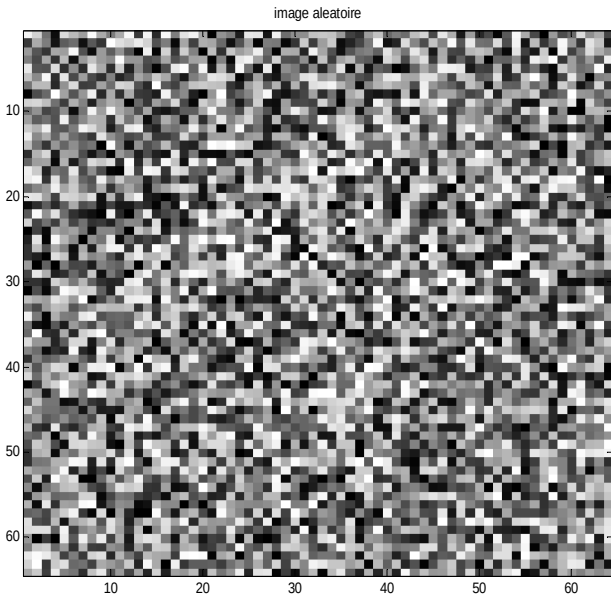
importantes). Nous allons donc utiliser la comparaison des coordonnées pour l'identification des images dans notre interface.

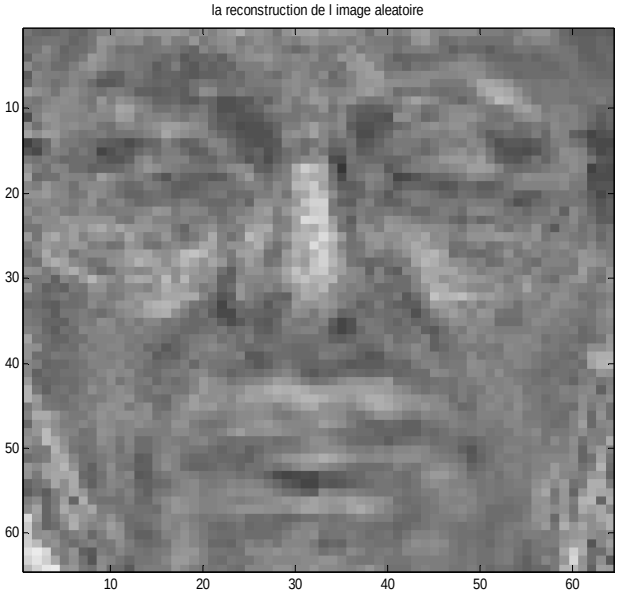
Pour terminer cette partie, on va tracer la courbe du nombre de visages reconnus en fonction de K, afin d'estimer la valeur optimale pour K. La courbe obtenue stipule qu'une valeur de K=18 conviendrait parfaitement tant au niveau performance de reconnaissance qu'au niveau d'efficacité en temps de calcul.



CLASSIFICATION VISAGE/NON VISAGE:

Nous créons d'abord une image aléatoire, puis on essaie de la reconstruire et affiche le résultat, enfin on comparera la valeur de son erreur de reconstruction avec celle d'un visage.



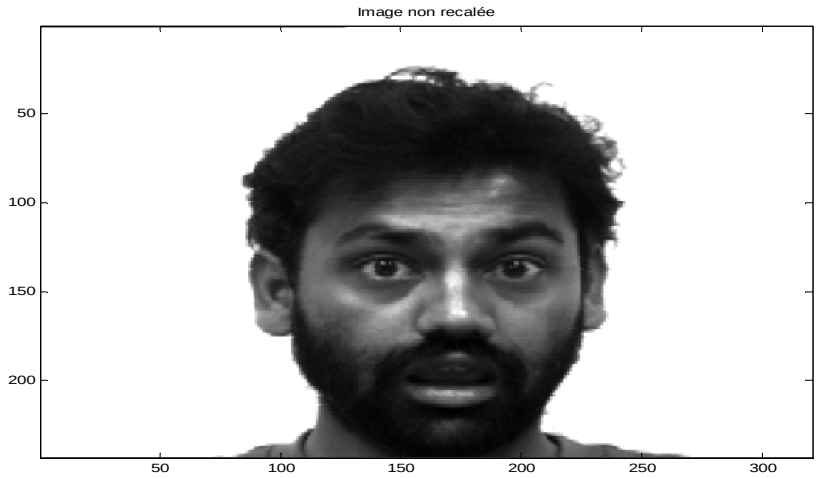


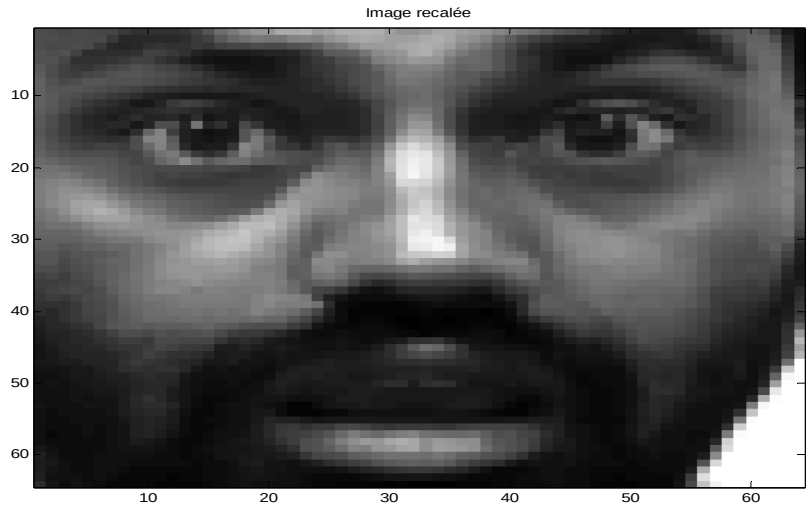
La reconstruction de l'image aléatoire donne un visage qui n'a rien à voir....Donc quand l'image est trop différente des visages, on a des reconstructions qui ressemblent à la base. On a une erreur minimale de 32 alors qu'avec des images on avait une erreur inférieure à 8 pour K>=30. On pourra donc différencier facilement un visage d'un non visage lors de l'implémentation de l'interface finale.

RECALAGE DES VISAGES:

Jusqu'à maintenant, nous avons travaillé sur une base d'images qui ont été préalablement préparées avec soin, en effet, ce sont toutes des images de visages recalées. Il est intéressant de comprendre comment s'effectue ce travail de recalage. C'est exactement cela le but de cette partie.

Nous allons implémenter une interface à travers laquelle l'utilisateur peut sélectionner un visage et effectuer ensuite le recalage. Notons que le recalage peut être effectué à priori à partir de n'importe quelle partie du visage, mais dans notre étude, nous allons procéder à un recalage par rapport aux yeux.





INTERFACE COMPLETE

Les calculs préalables nous permettent d'étudier le système pour pouvoir fixer tous les paramètres nécessaires à une identification d'une image test. En effet, avec nos images de référence, on prendra 30 eigenfaces pour projeter les images à identifier. Ce qui donne une erreur de reconstruction faible pour toutes les images qui ressembleraient à un visage de la base. De plus, si l'erreur de reconstruction est supérieure à 10, on considérera que l'image est un non visage. Et pour une erreur d'identification supérieure à 8, on considérera que l'image est un visage mais n'est pas reconnu. Ce qui nous permet d'implémenter une interface capable de savoir si une image est un visage ou non et s'il est reconnu. L'interface permet de récupérer en un premier temps les images *subject(x)_cat(y).jpg* du répertoire *recale* que l'on charge dans une matrice I (pour chaque catégorie on charge les 15 photos de celle-ci et à la suite on passe à la catégorie suivante) . Grâce à l'analyse en composantes principales, on récupère les eigenfaces formant la base du sous-espace de l'ensemble des visages. Celles-ci nous permettront de projeter l'image test dans ce sous-espace.

Ensuite, l'interface permet de choisir une image quelconque et de la recaler manuellement. Pour cela, on fait appel à la fonction **recalaginter ()** (qui elle-même appelle la fonction **recalage ()**) qui nous renvoie l'image recalée et l'image d'origine, les 2 images étant préalablement converties en noir et blanc. On affiche ces 2 images puis nous allons étudier la projection de l'image recalée dans le sous-espace. Le seuil d'identification est fixé à 7 et le nombre d'eigenfaces pour la projection à 30. Enfin, le seuil de reconstruction est fixé à 12.

Après, nous récupérons les coordonnées dans la base des 30 eigenfaces par un produit scalaire. Ce qui donne l'image test projetée dans le sous-espace. Et l'erreur de reconstruction et l'erreur d'identification sont calculées. Un message est affiché selon si l'image est un visage ou non et si ce visage est reconnu.

Enfin, le bilan de l'étude de l'image test est affiché, à savoir :

- l'erreur de reconstruction
- la classification visage/non-visage de l'image
- l'erreur d'identification
- une figure avec l'image reconstruite
- une figure avec le visage le plus proche de la base si c'est un visage
- une figure avec le visage de la personne si elle est reconnue

Si nous voulions étudier une autre base de données de visages, il suffirait de changer le début pour récupérer une matrice I qui contient toutes les nouvelles images. Ensuite il faudrait changer les valeurs de m et n selon le nombre d'images et la taille de ces images. Après, on relancerait les programmes réalisés pendant l'étude pour avoir les paramètres optimaux K, θ et T. Enfin, on peut changer les valeurs du vecteur qui fixe les paramètres de recalage pour pouvoir centrer l'image par rapport aux lèvres etc.… A part ça, le programme continuerait à fonctionner. C'est donc un bon squelette qui peut servir à l'étude de tout ensemble de visages.

BILAN

Ainsi, nous avons réussi à simuler un système de reconnaissance de visages à partir d'une certaine base de données de visages. Nous avons pour cela fait une décomposition en composantes principales et réaliser des expériences pour déterminer les paramètres idéaux pour les reconnaissance. Ce processus d'identification constitue la dernière étape d'une chaîne où interviennent du matériel et des logiciels.

Nous allons nous intéresser par exemple, à un système de sécurité dans un site sensible (centrale nucléaire) qui permettrait d'identifier le personnel qui entre dans certaines salles en scannant le visage de

la personne. Pour réussir à mettre en place ce système, il faudrait d'abord récupérer l'image de la personne à identifier en prenant la personne en photo de face. Ceci se fait grâce à un appareil photo numérique attaché à une unité mobile pour s'ajuster au niveau du visage. Cet appareil serait relié à l'ordinateur ou un logiciel récupérerait l'image. On aurait comme logiciel un programme ayant 2 modes, le mode administrateur pour rajouter des nouveaux visages à la base de donnée et le mode utilisateur pour identifier. Dans le premier mode, le programme aura une interface qui nous permette de rajouter une image en la sélectionnant et en la recalant manuellement (comme dans notre interface). De plus, il faudrait que tous les paramètres soient actualisés (Imoy, K, T..) grâce à une implémentation qui relancerait les petits test et récupèrerait les bonnes valeurs. Après, dans le deuxième mode, l'image serait déjà recalée (en plaçant bien l'appareil grâce à des capteurs optiques…) et donc on procèderait à son ACP avec l'interface que l'on a implémenté. Mais bon on évitera de dire à la personne que ce n'est pas un visage en cas d'erreur et se contentera du même signal sonore avec une lumière rouge que pour une personne non autorisée. Et il faudrait mettre en mémoire un moment les visages qui n'appartiennent pas à la base pour que si cette personne essaye de rentrer plus de 3 fois, l'alarme soit déclenchée.

Ce système présente aussi des limites car si l'expression du visage au moment de la photo est trop différente de celles qui sont présentes dans la base l'interface ne le reconnaît pas. En plus, si on est dans une grande entreprise la rentrée de toutes les données est assez pénible. Enfin, il vaut mieux un appareil photo fixe et un recalage manuel pour ne pas supprimer un poste. Déjà qu'à cause des digicodes il n'y a plus de concierges !!!

Annexe: Explication de tous les codes et leur utilisation

MI0: Ce programme sert de point de départ, c'est le premier programme à lancer, en effet il initialise toutes les donnés qu'on aura besoin pour la suite: tout d'abord la matrice I et J qui contiennent respectivement les visages de références et tous les visages (à savoir référence et test), puis calcule et affiche l'image moyen Imoy, enfin il génère les matrices A et AA qui représentent les images centrées des visages de référence et de test.
ATTETION: MI0 doit être exécuté en premier pour pouvoir faire fonctionner les autres programmes!

MI1: Il affiche simplement les cinq premières eigenfaces avec leur valeur propre.

MI2: On commence ici la partie 'Projection dans le sous espace des visages', cette première partie calcule les coordonnées d'un visage dans le sous espace et on affiche quelques visages avec leur visage reconstruite.

MI3: Après avoir normalisé les valeurs propres, on trace la courbe de leur somme cumulée en fonction de K.

MI4: Enfin, on illustre l'évolution de la moyenne de l'erreur de reconstruction des visages de test pour K variant de 1 à m. On doit retrouver une certaine cohérence avec le résultat précédent.

MI5, MI6, MI7: Ces trois programmes sont consacrés à la tâche d'identification. On va notamment illustrer la distance de chaque visage de référence par rapport à tous les autres par une image de matrice puis comparer deux méthodes de comparaison et enfin sortir la courbe du nombre de visages reconnus en fonction de K.

MI8, MI9: On entame la partie classification visage/non visage, on va d'abord étudier les caractéristiques d'une image quelconque, effectuer sa reconstruction pour comparer avec celle pour un visage. Puis nous allons effectuer la classification visage/non-visage à partir d'une base de données quelconque.

MI11: Cette partie nous permettra d'effectuer le recalage manuel d’une image à travers une interface.

Fonctions utilisées:

categories: fonction faisant la correspondance entre un entier et une catégorie de visage, par exemple 'happy' ou encore 'wink'.

showface, showcface: fonctions permettant d'afficher des visages (centrés ou non).

recalage, recalaginter: fonctions permettant le recalage des images.